Implementing Yasuyuki's FPHX silicon tracker in simulations with ladders

Tony Frawley

August 18, 2015

The thickness budget with FPHX chip

The change from the SVX4 chip to the FPHX chip for reading out the silicon strip tracking layers results in the need for much less cooling. The material budget can be reduced to the following.

Layer	Radius	X0	
P0	2.5cm	1.3%	
P1	5.0cm	1.3%	
S0	8.0cm	2% radiation length	(For double layer S0a+S0b)
S1	32 cm	1.2%	(For double layer S1a+S1b)
S2	56 cm	1%	

I made some estimates of the mass resolution, using this thickness budget, to try to see what the layer radii should be to achieve the needed 100 MeV Upsilon mass resolution.

These estimates are made in the cylinder cell geometry. In the cylinder cell geometry:

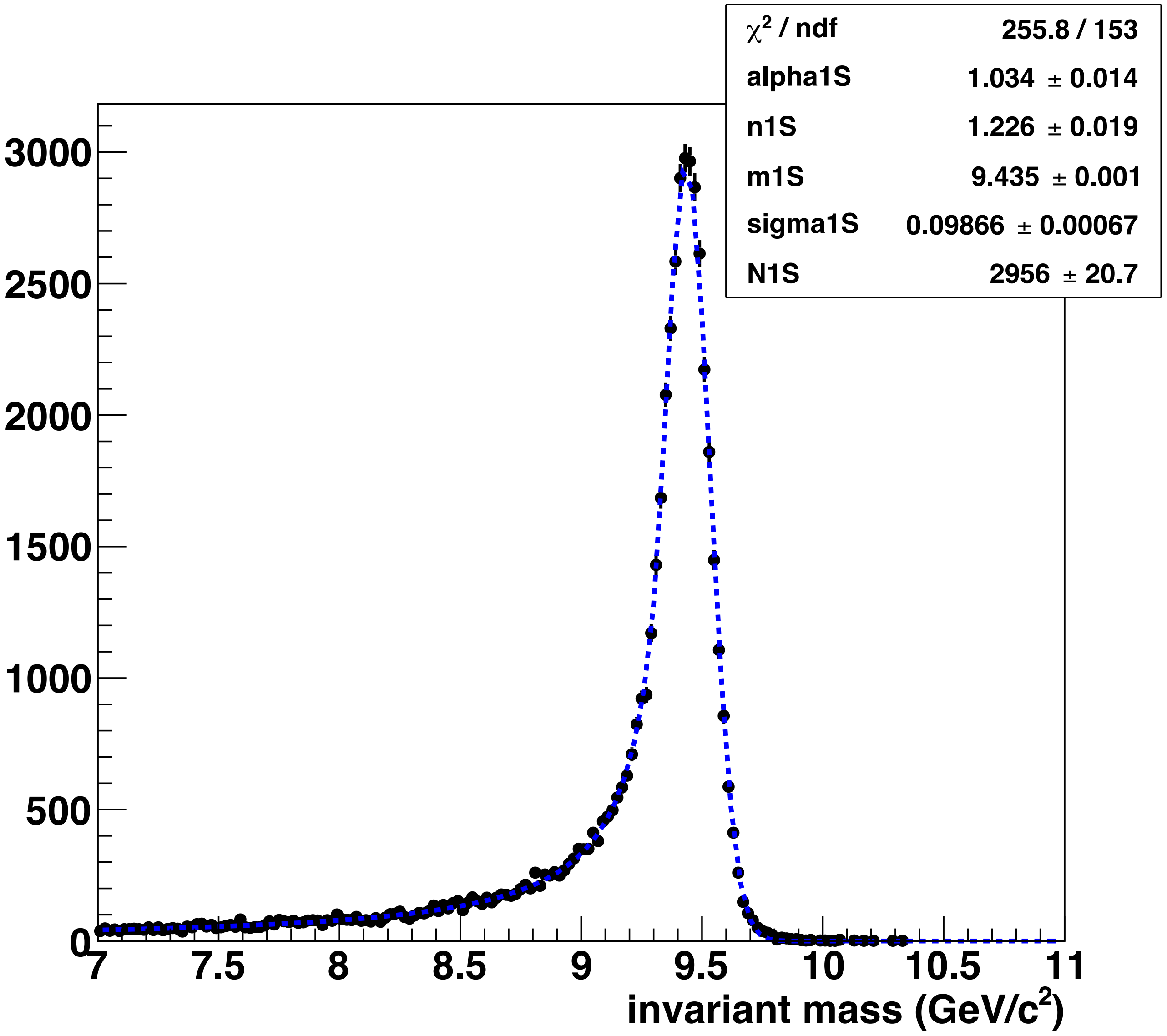
- All sensors are in the same layer
- There is no gap between sensors in phi
- The thickness of the electronics is represented by a layer behind the sensor

In real life, this is not possible. Real life will make the thickness larger.
More on that later.

Results with the cylinder cell model (64 cm outer radius)
uses 2 layers of PHENIX pixels

Layer	Radius (cm)	cell size (cm x cm)	X0 (%)
P0	2.7	0.005 x 0.0425	1.3
P1	4.6	0.005 x 0.0425	1.3
S0a	7.5	0.0058 x 9.6	1.0
S0b	8.5	0.0058 x 9.6	1.0
S1a	35.0	0.0058 x 9.6	0.6
S1b	37.0	0.0058 x 9.6	0.6
S2	64.0	0.0060 x 9.6	1.0

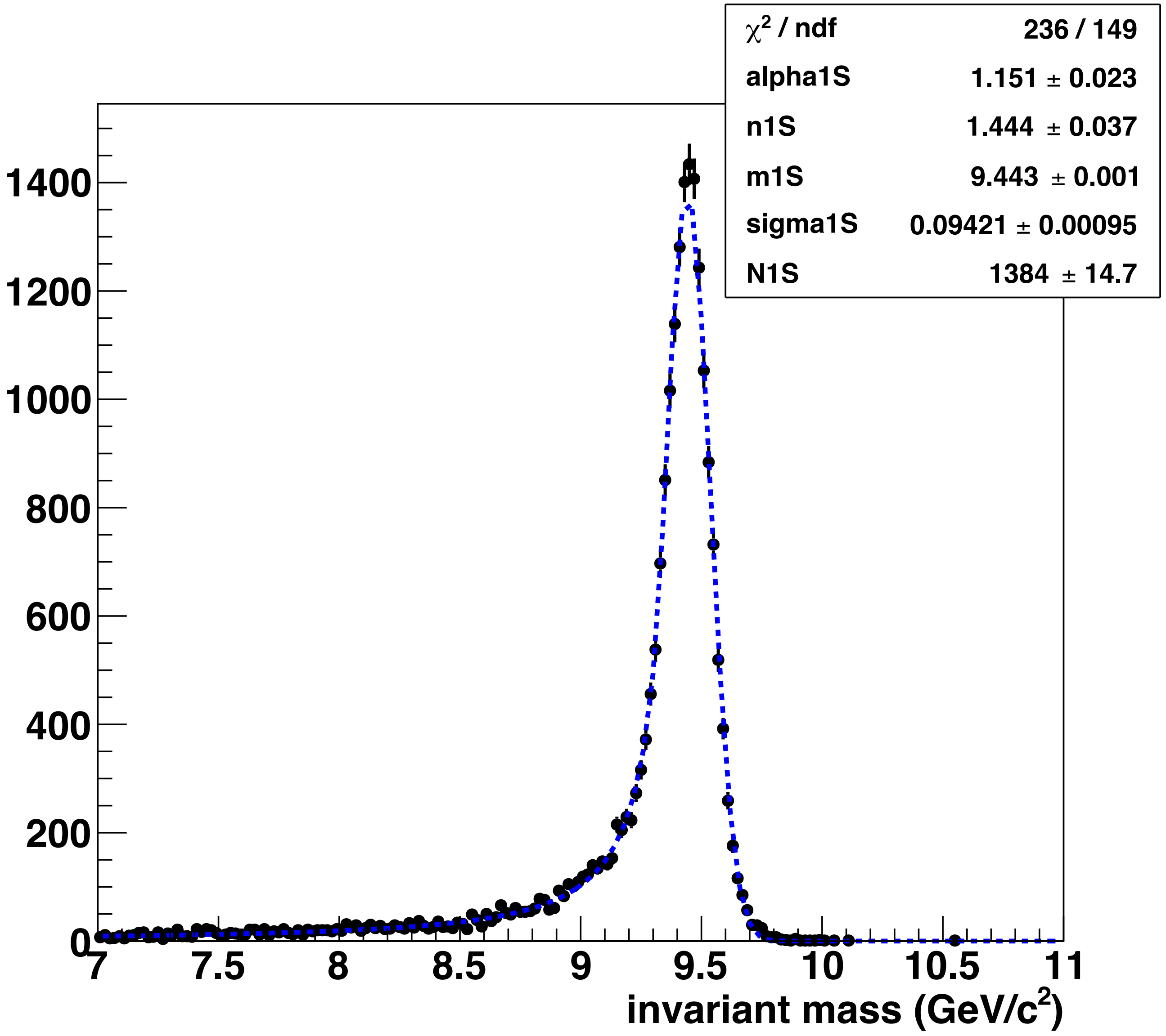
mass resolution 98.7 MeV +/- 0.7 MeV



Same cylinder cell model for outer tracker
use ITS inner pixels instead of PHENIX pixels

Layer	Radius (cm)	cell size (cm x cm)	X0 (%)
P0	2.3	0.002 x 0.002	0.3
P1	3.2	0.002 x 0.002	0.3
P3	3.9	0.002 x 0.002	0.3
S0a	7.5	0.0058 x 9.6	1.0
S0b	8.5	0.0058 x 9.6	1.0
S1a	35.0	0.0058 x 9.6	0.6
S1b	37.0	0.0058 x 9.6	0.6
S2	64.0	0.0060 x 9.6	1.0

mass resolution 94.2 +/- 0.9 MeV



Creating a ladder model

Creating a realistic ladder model requires:

- A module to construct the geometry in G4
- A stepping module to extract energy deposited in the active detector volumes
- A hit object to store the hits
- A geometry object so that the hit positions can be found later

Ideally, the module that constructs the geometry in G4 should allow as much as possible of the configuration to be specified from the calling macro (i.e. not hard coded).

The ladder model

The silicon tracker ladder model is created in:

coresoftware/simulation/g4simulation/g4detector/
PHG4SiliconTrackerSubsystem
PHG4SiliconTrackerDetector

The hits are accumulated by:
PHG4SiliconTrackerSteppingAction

The hits are stored in:
PHG4Hitv1

The geometry is stored in:
coresoftware/simulation/g4simulation/g4detector/
PHG4CylinderGeomv4

The model was originally made to describe the revised MIE configuration, using the SVX4 chip.

I presented some results from this in May, where I found a mass resolution of about 110 MeV for the ladder equivalent of the revised MIE configuration (roughly 10% worse than for the cylinder cell model).

The ladder model (cont.)

Unlike the cylinder cell model, the ladder model requires that there be some **overlap** of the sensors in phi to avoid inefficiency for curving tracks. I was assuming 15% overlap in phi.

This necessarily makes the ladder geometry thicker on average than the cylinder cell geometry.

Ideally, one would like to accomplish the necessary overlaps of the sensor with two staggered radius values for each layer. For the SVX4 version this was not possible because the ROC + FPGA dimension in the phi direction was too large. It caused overlaps in phi between adjacent ladders at the same radius.

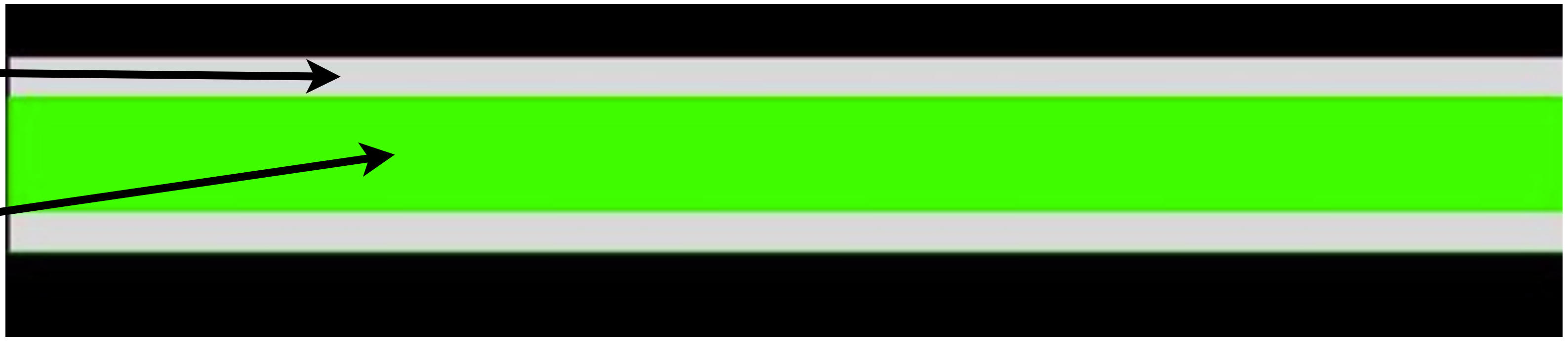
Therefore it was necessary to use:

- 4 staggered radii for layer S0
- 3 staggered radii for layer S1
- 2 staggered radii for layer S2

For the double layers S0 and S1, there is one stave with a sensor on the inside and out.

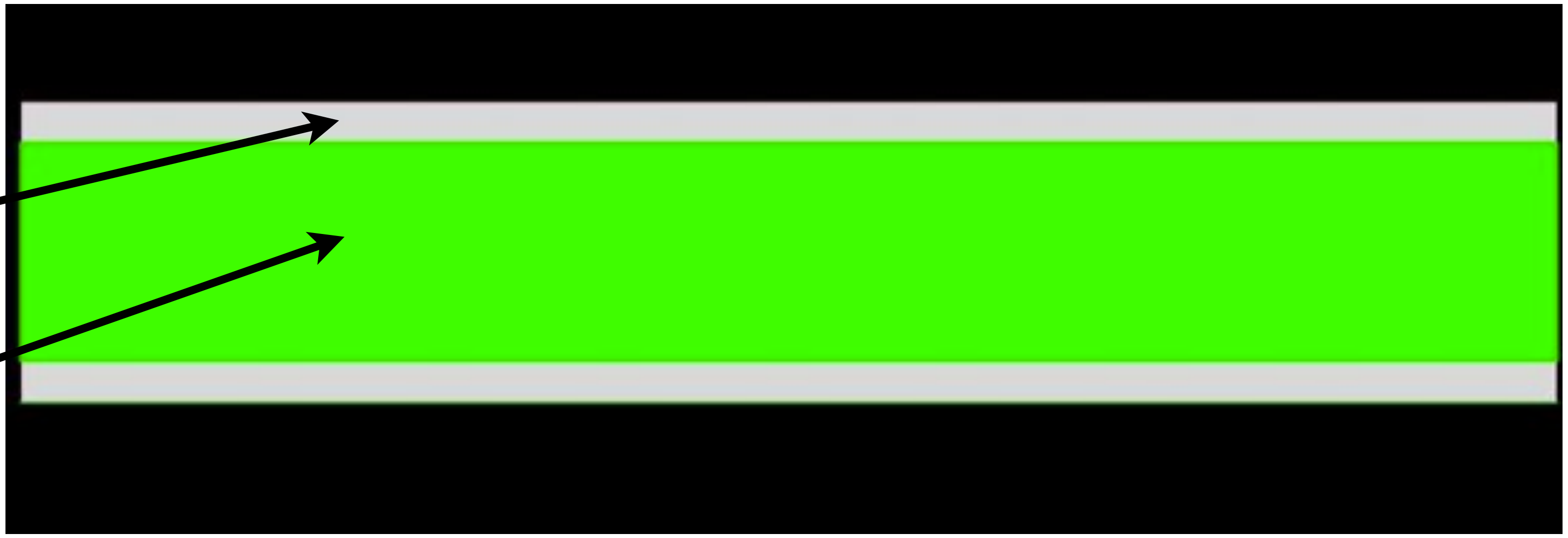
I will show some snapshots of the components of the ladders from screenshots of G4 displays in the next few slides.

Components - sensor



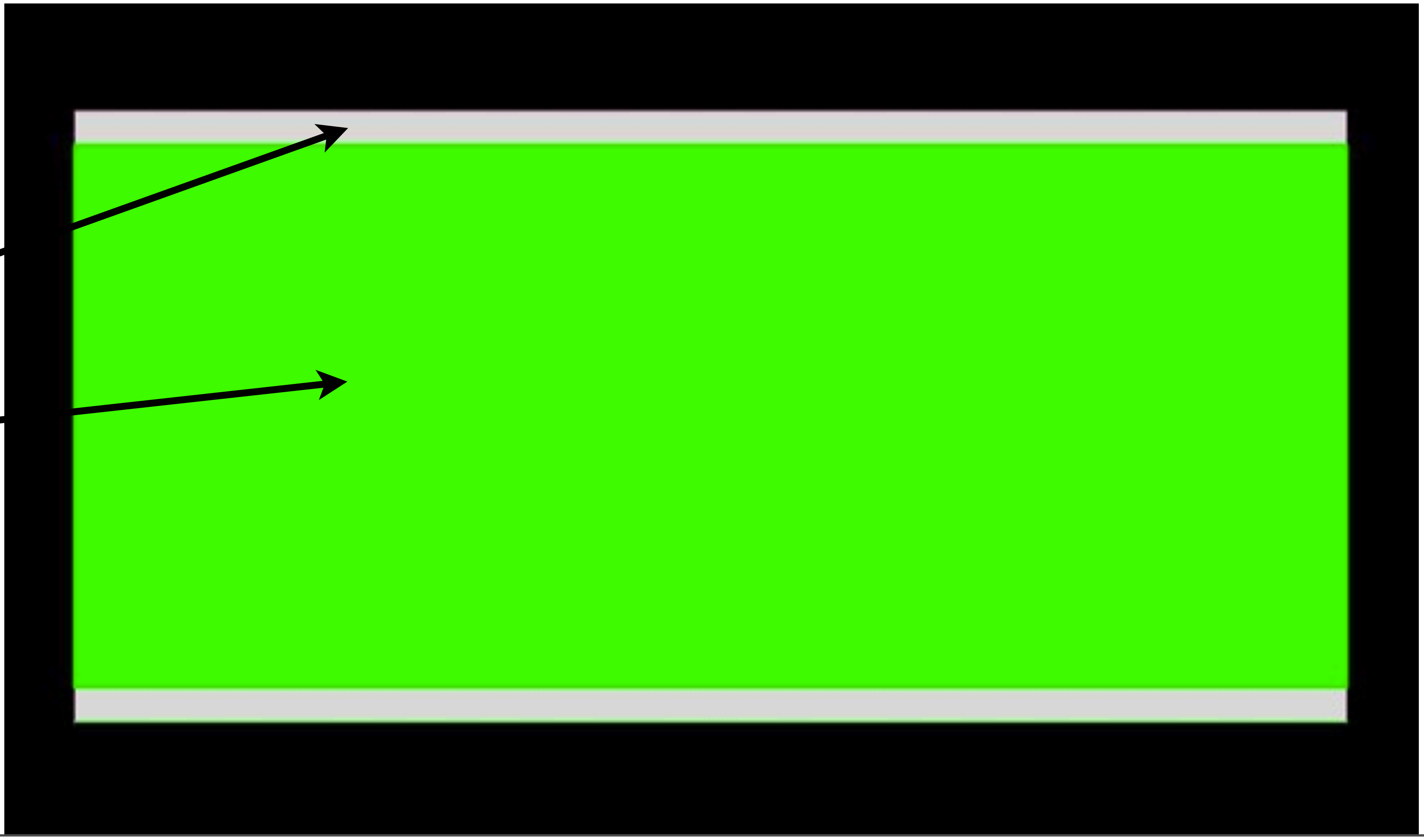
Sensor inactive edge

Sensor strip area
128 strips in Φ



Sensor inactive edge

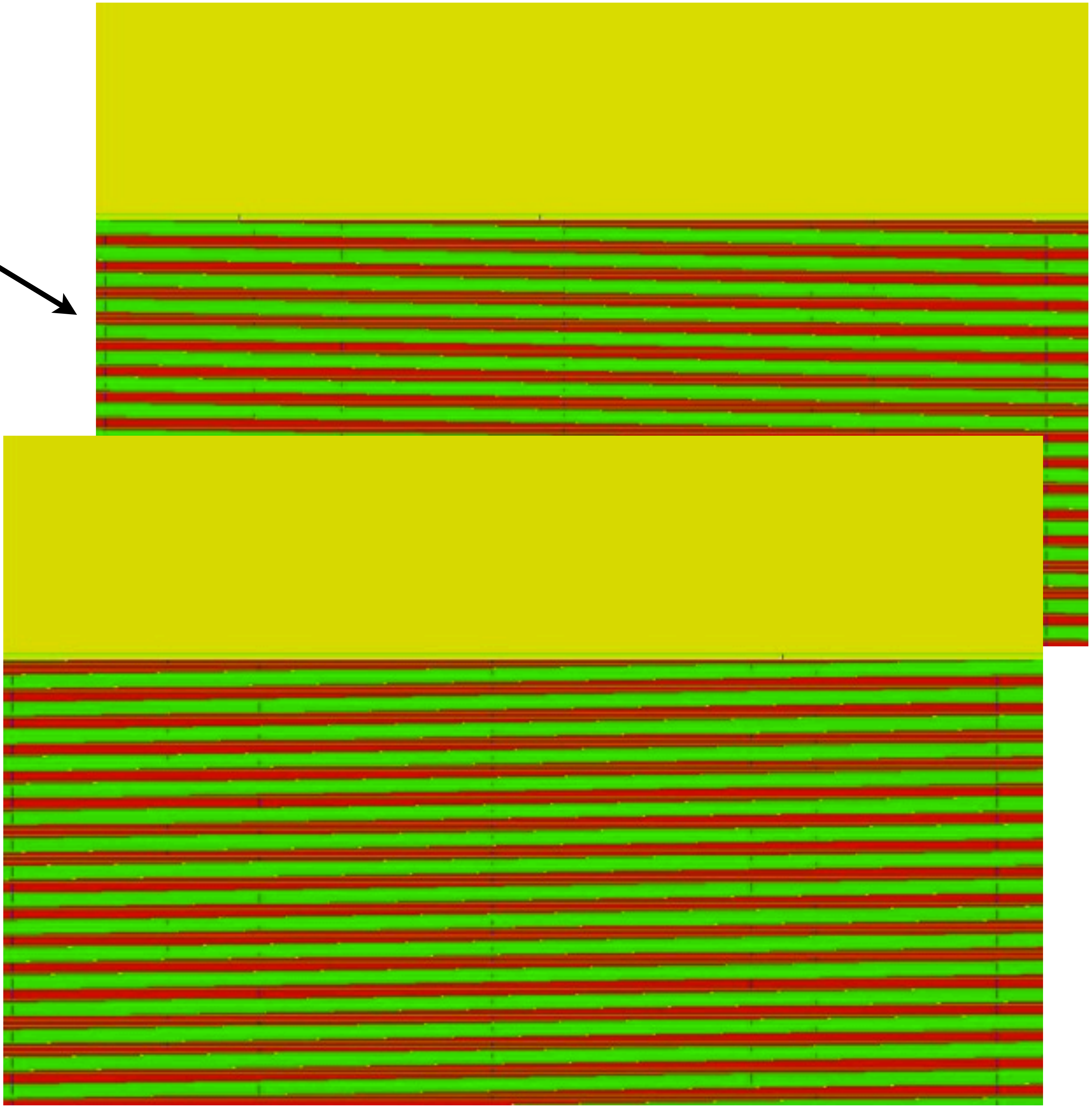
Sensor strip area
256 strips in Φ



Sensor inactive edge

Sensor strip area
768 strips in Φ

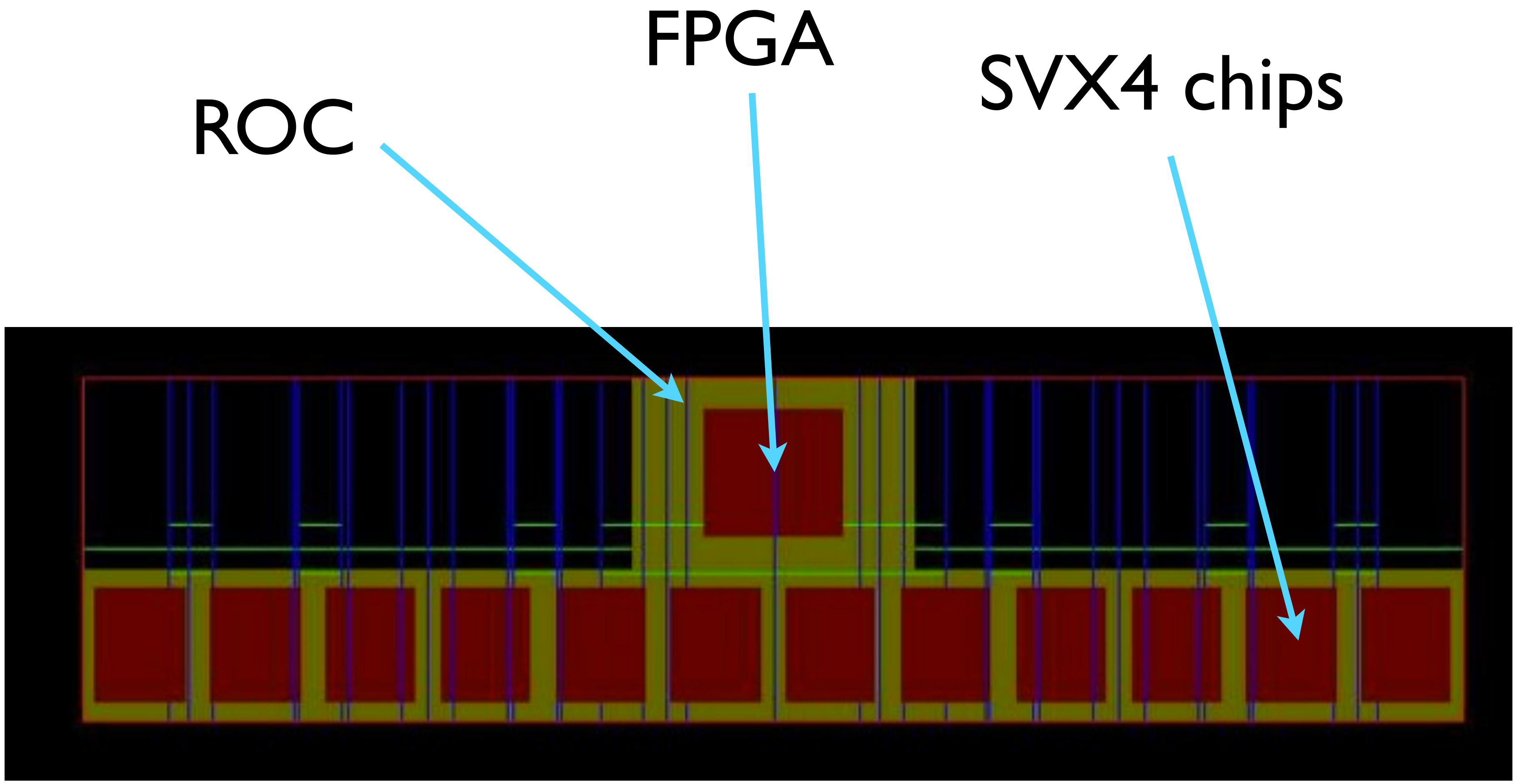
Components - sensor



Strip tilt +1/64

Strip tilt -1/64

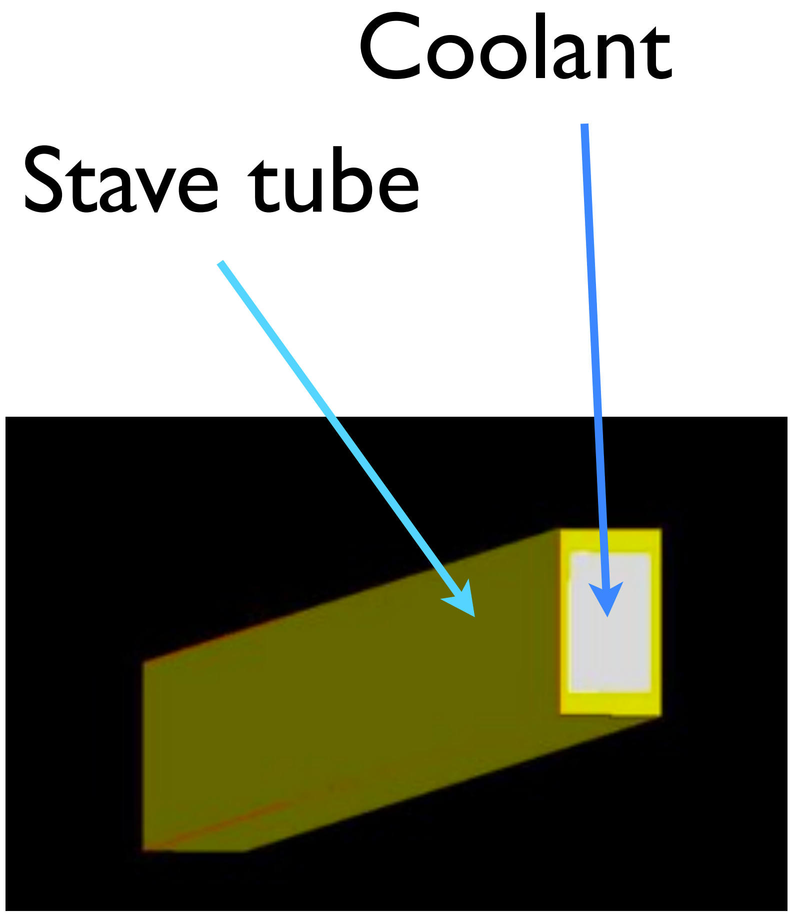
Components - ROC, SVX4 chips, FPGA and stave



ROC

FPGA

SVX4 chips



Stave tube

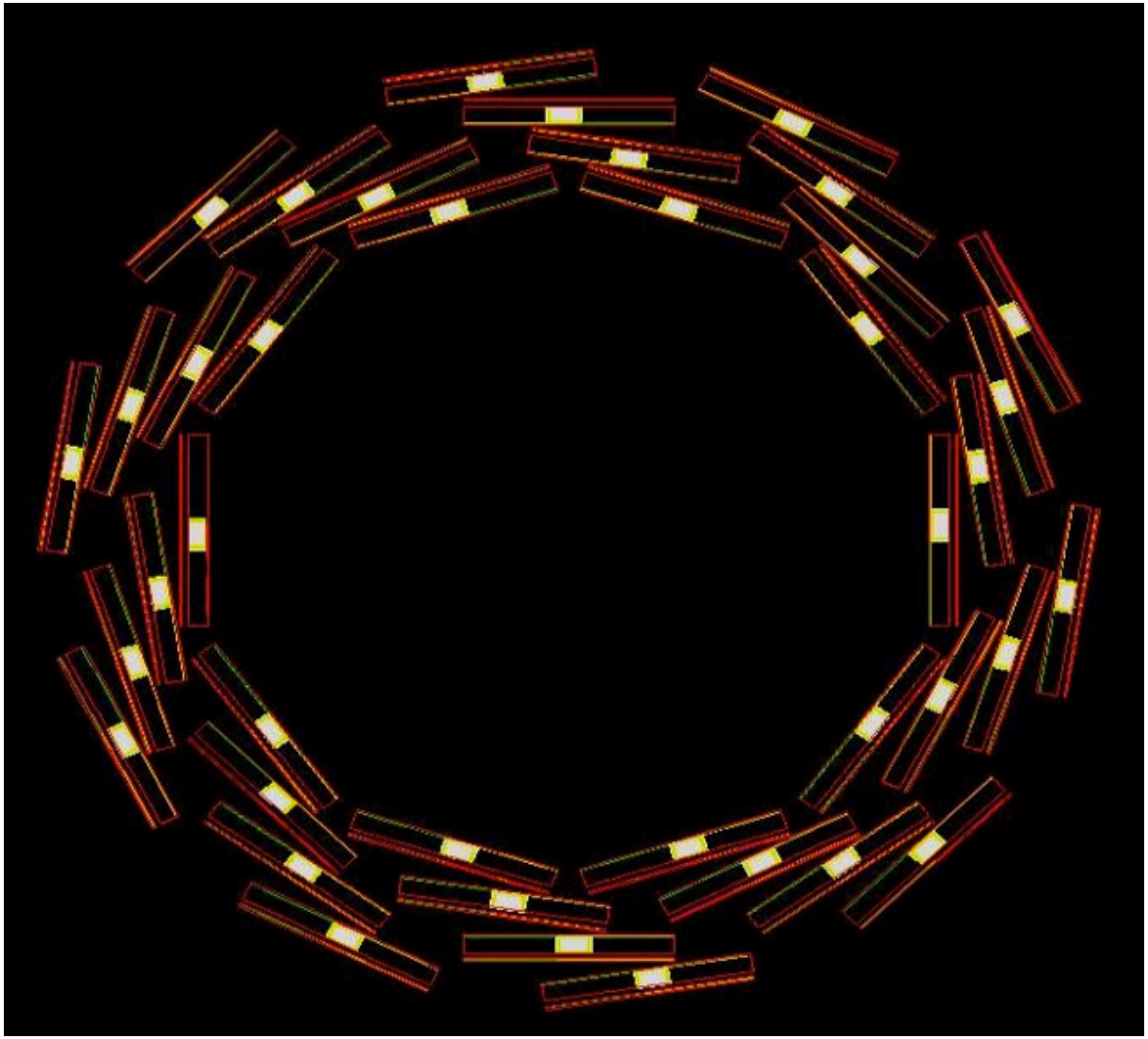
Coolant

Layers 0 and 1 (8.5 cm inner radius, 256 strips in Φ)

16 mm strips - all strips read out

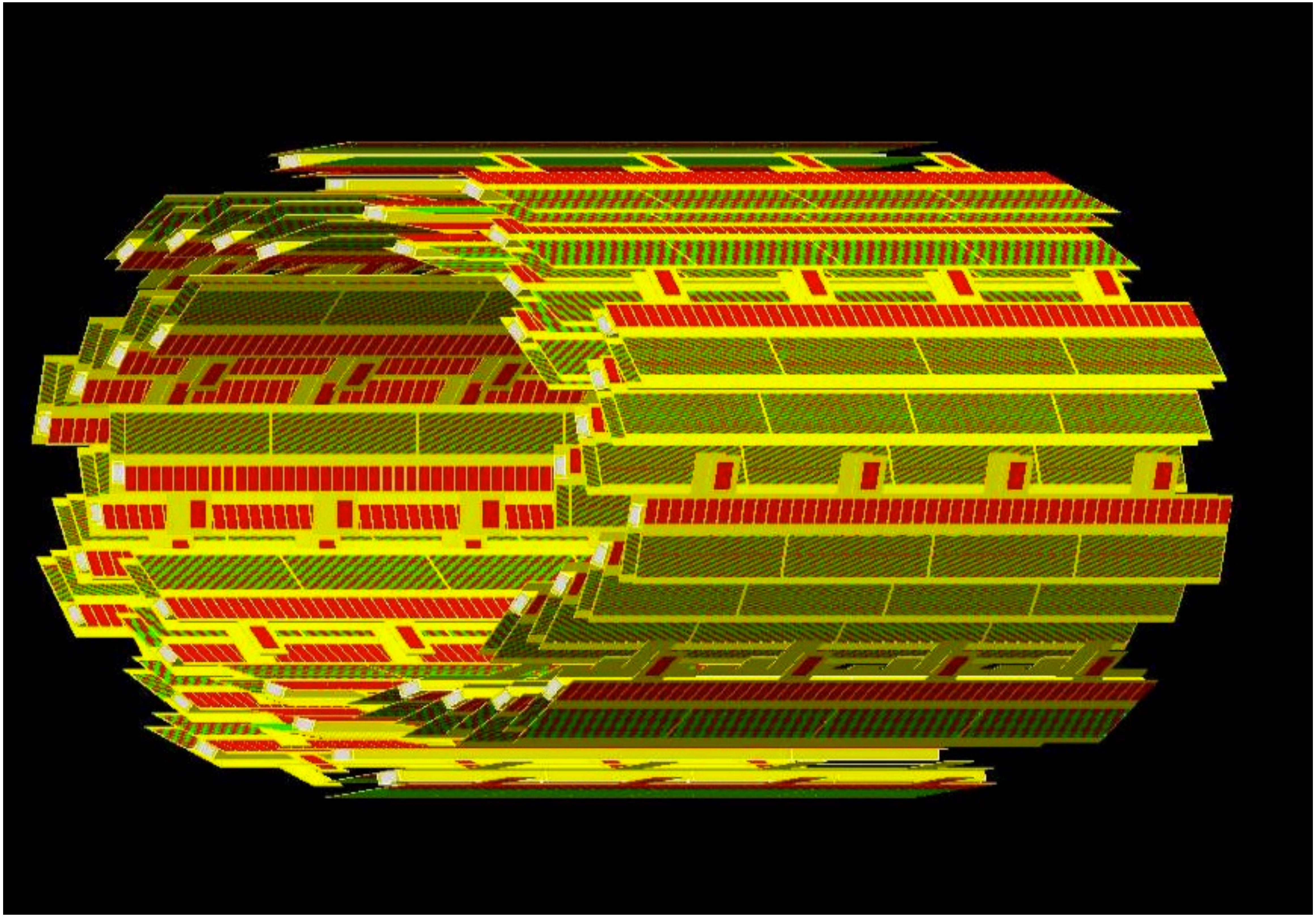
ROC and FPGA on one side only of sensor

otherwise need too many staggered radii!



SVX4 version: Layers 0 and 1 (8.5 cm inner radius, 256 strips in Φ)

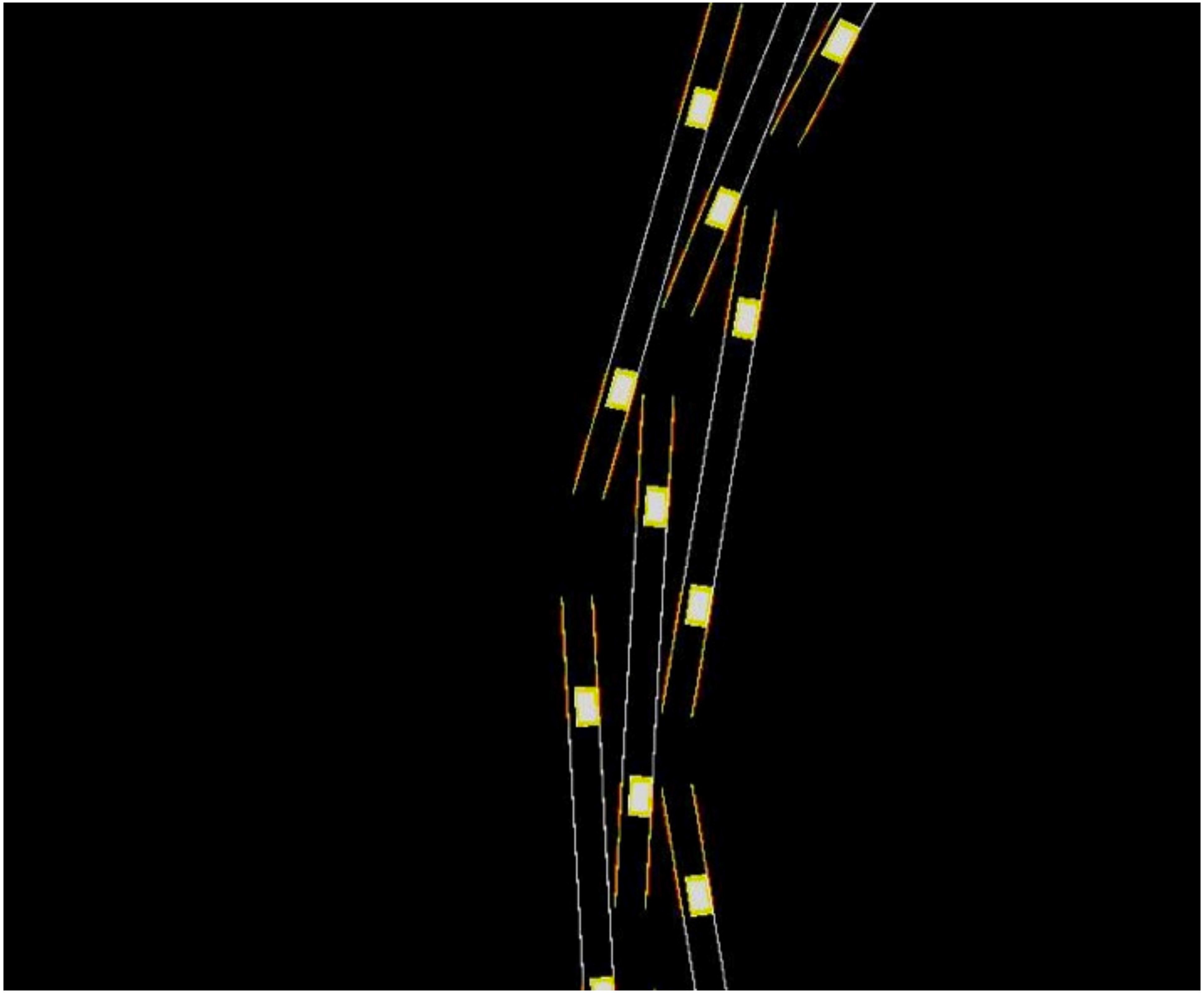
16 mm strips - all strips read out



Layer 2, 3 (40 cm inner radius, 768 strips in Φ)

8 mm strips - 3 strips connected together

ROC and FPGA on both sides of sensor

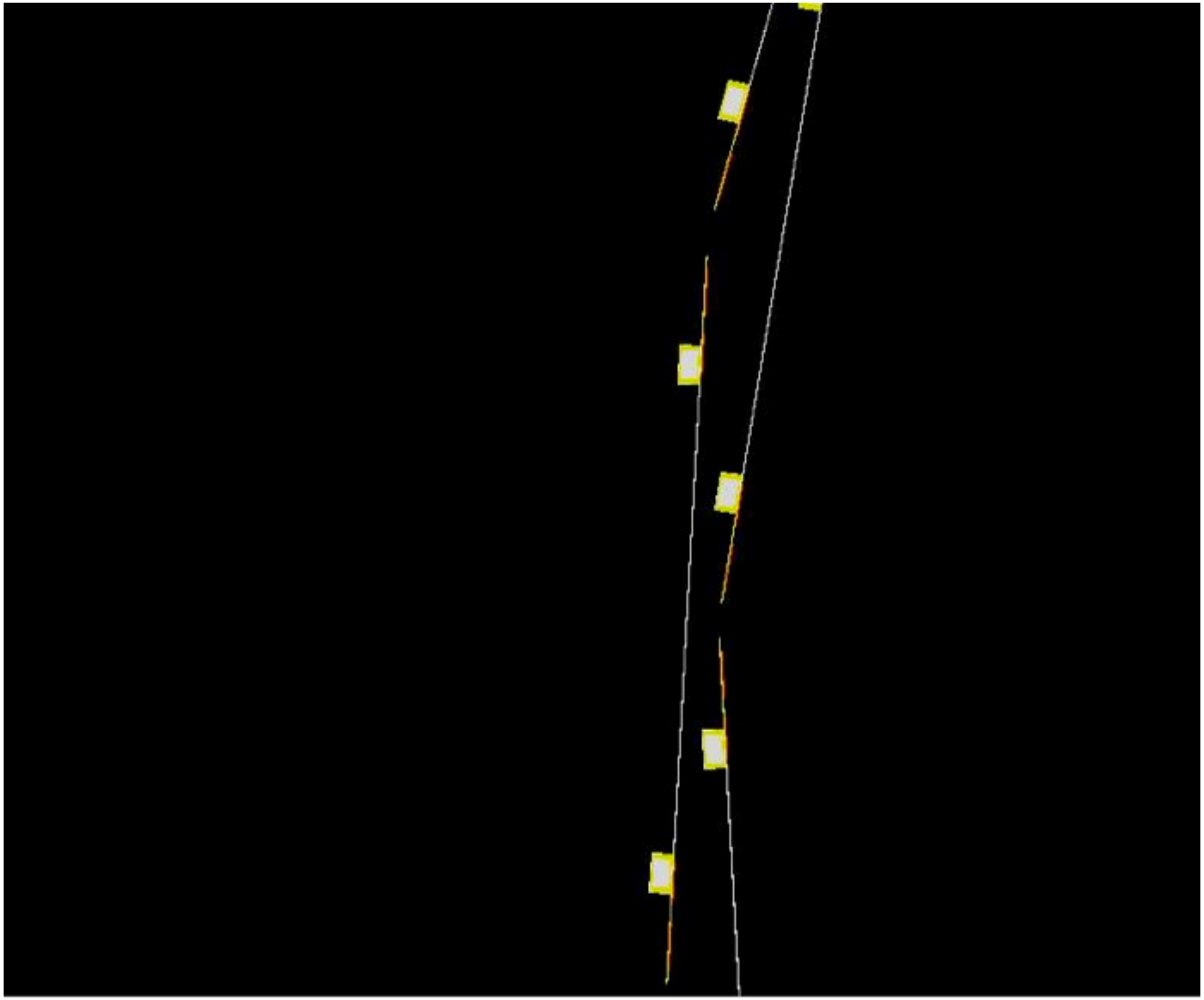


Layer 4 (80 cm inner radius, 1536 strips in Φ)

8 mm strips - 6 strips connected together

16 mm strips - 3 strips connected together

ROC and FPGA on both sides of sensor



Code reorganization for FPHX chip version

The ladder model code was written in such a way that it would have been messy to change it to describe Yasuyuki's proposed FPHX configuration.

- The cooling and support structure is very different
- There is no longer an FPGA

Changes to:

coresoftware/simulation/g4simulation/g4detector/
PHG4SiliconTrackerDetector
PHG4SiliconTrackerSubsystem

Changes:

I have just finished a reorganization of the code to make it modular, so that parts of the ladder can easily be switched off (FPGA), or substituted with a different geometry (support and cooling).

At the same time, I made it possible to completely define the geometry of the detector from the calling macro (G4_Svtx_ladders.C), including switching between SVX4 and FPHX configurations. There are now no hardcoded dimensions at all.

I am now ready to write a new module describing the FPHX support structure - this is the only task remaining.

SVX4 chip ladder model

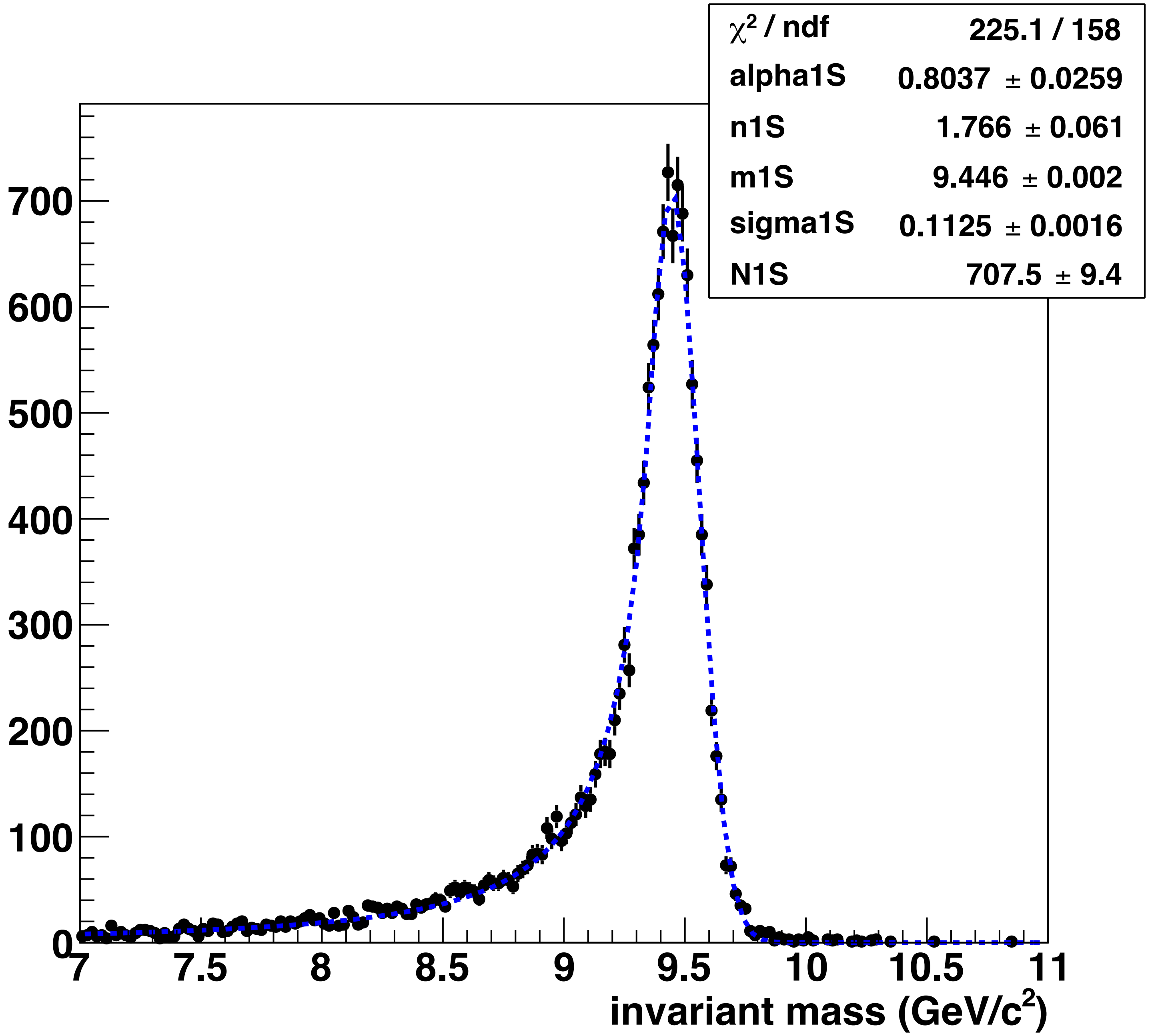
This is a calculation with the reorganized ladder construction code, for the SVX4 ladder version with the revised MIE radii.

This gives a resolution of 112 MeV.

The sensor overlap in phi is 15%.

Note however that the tracking code has not been re-optimized following some extensive rewrites, so this may not be the final word.

Mike McCumber is working on the re-optimization.



What next?

Implement the new cooling and support structure. Should take a couple of days.

Optimize the tracking for the new configuration (tweak parameters).

Determine what the radii of the tracking layers really have to be, since the thickness will increase.

Study and optimize the pattern recognition in Hijing events. Issues are:

- We cannot read out every strip, so some strips will be connected together.
- The strips are 8 mm long - do we need some stereo strips?

Settle on a configuration that we can defend before the pCDR.